REMARKS

The new rejection of claims 26-33 and 42 under 35 USC §102(e) in view of U.S. Patent Publication No. 2002/012478 (Kocher) is respectfully traversed on the grounds that the Kocher publication fails to disclose or suggest the claimed combination of:

- falsifying input data by combination with auxiliary data before the execution of one or more operations;
- combining the output data determined by execution of the one or more operations with an auxiliary function value in order to compensate for the falsification of the input data; and
- the auxiliary function value having been previously determined by the execution of the one or more operations with the auxiliary data as input data in safe surroundings and stored along with the auxiliary data.

The Kocher publication discloses the first two steps, but not in combination with **pre-determination** in **safe surroundings** and **storage** of the auxiliary function value and auxiliary data, as claimed. Instead, Kocher discloses that the auxiliary data and auxiliary function value are computed **while** computing the permutation of the input data. As a result, the auxiliary function value and data are vulnerable to inspection by an attacker.

The present invention provides a way to prevent exposure of secret information even if an attacker is able to intercept signal patterns resulting from execution of operations involving the secret data. This involves falsifying secret input data with auxiliary data before execution of the operation(s), and then combining the output with auxiliary function values so that the result is the same as it would have been if the original secret data had been operated on without falsification. This procedure is also disclosed by Kocher.

However, a problem with falsification of input data using auxiliary data and auxiliary function values is that it might be possible for an attacker to determine the auxiliary data and auxiliary function values as they are generated. This problem is not even considered by Kocher, but is solved in the claimed invention by using auxiliary data and auxiliary function values that have been previously calculated in safe surroundings, and stored for use in connection with the input data falsification.

Claim 26 specifically recites this previous determination of the auxiliary data and function values in paragraph 3 of the body of the claim, as follows:

- *wherein the auxiliary function value (f(Z)) was **previously** determined by **execution of the one or more operations (f) with the auxiliary data (Z)** as input data **in safe surroundings** and stored along with the auxiliary data (Z).*

This clause specifically uses the words "previously" in "safe surroundings" requiring determination of the auxiliary function value to be carried out before the method comprising the falsifying and combining steps. While the claim does not preclude the falsifying and combining steps from also being carried out in safe surroundings, the point of the invention is that those steps could be carried out in unsafe surroundings without compromising data, so long as the previous determination of the auxiliary function value is carried out in safe surroundings.

The Kocher publication also discloses protection of secret data against external attacks. One of the techniques disclosed by Kocher is a special method for implementing a permutation. An array dataIn of boolean values is given as input data and an array table of integer values specifies a given permutation of n numbers 0 to n - 1, where n is the length of the array dataIn. As explained in paragraphs [0066] and [0067] of Kocher, this "input-ordered permutation" involves correlating an array dataOut to dataIn[i] for every I from 0 to n - 1. However, as explained in paragraph [0068] of Kocher, a further step of blinding the input data is used to protect the computation of the permutation against an attack on the input data. This involves blinding the input data by random bits and the performing the permutation of the blinded input data with the help of an additional, randomly generated permutation (which corresponds to the claimed auxiliary function).

The exact method of blinding used by Kocher is given in the form of source code included in paragraph [0068], as follows:

- Suppose the length of the input array dataIn is 64. An array *perm* holding the numbers from 0 to 63 in a randomly permuted manner is computed in a first step (see the first and second for-loops of the listed source code).
- In a second step, another array *temp* of length 64 is computed by setting for every I from 0 to 63:

  temp [p] := dataIn [p] ^ b,

  where ^ denotes the XOR operator,

  b is a random bit just computed, and

  p = perm [I], *i.e*, p is an index given by the previously computed random array perm.
- In a third step (also in the second for-loop), dataOut is temporarily set to dataOut [table [p]] = b (see the third for-loop of the listed source code).
- In a fourth step, the array perm is recomputed, *i.e.*, randomly permuted once more (see the fourth for-loop), although the general algorithm would work without this step.
- Finally, in the last step, the input data dataIn is finally permuted by setting, for every I from 0 to 63, where again p = perm[i]:

  dataOut [table [p]] := temp [p], which is the same as

  dataOut [table [p]) := dataIn[p] ^ b (see the second step above).

The blinding of the input data by the random bit b is compensated for by setting:

dataOut [table [p]] ^:= temp [p], which is a shorthand notation for

dataOut [table[p]] := dataOut [table[p]] ^ temp [p].

Summarizing, one obtains from the temp [p] correlations and the expression dataOut [table [p]] = b obtained in the third step described above:

dataOut [table[p]] := b ^ dataIn[p] ^ b = dataIn[p], which is what was originally to be computed since b ^ b = 0.

It can be seen from this analysis of the blinding method listed in paragraph [0068] of Kocher that only the order in which the operations dataOut [table [p]] := dataIn [p] were performed is permuted by the random permutation*perm*, by setting p = perm [I] where I runs from 0 to 63. Furthermore, the input bits dataIn [p], where temporarily blinded by the random bits b are compensated for after the permutation of the blinded input bits.

In particular, the first step of the claimed method, namely falsifying of input data (dataIN), is carried out by combination with auxiliary data in the form of random bits b before the execution of one or more operations (the permutation of the array dataIN according to the order given by the source code). The second step of the claimed method, namely compensation for the falsification, is carried out by determining the output data by the execution of one or more operations (dataIn [p] ^ b = temp [p]) combined using dataOut [table [b] ^= temp [p] with an auxiliary function value (dataOut [table [b]] = b). However, the third step, namely computing the auxiliary data (random bits b) and auxiliary function value (the permutation of b temporarily stored as dataOut [table [p]]), *is carried out* **while** *computing the permutation of the input data.* **In other words, in contrast to the present invention, the auxiliary data and the auxiliary function value are not previous computed in safe surroundings, as claimed, but rather are computed internally and thus may relatively easily be inspected by an attacker upon analysis of signal patterns reflective of the auxiliary data and function value computation.**

The claimed storage in safe surroundings is an important aspect of the invention and in fact was explicitly described as such in the original specification. As explained at the end of the second paragraph on page 3 of the present application:

> . . .It is important in this context for the random number and the function value [*i.e.*, the auxiliary data and the auxiliary function value] to be previously determined and stored in safe surroundings so that the calculation of the function value from the random number cannot be intercepted . . .."

Furthermore, as explained in line 7 on page 7 of the original specification, this previous determination could be during production of the card or, more precisely (as explained in the last line of the third paragraph on page 8), during the personalization phase of card production. In fact, the original specification discusses what happens when non-stored values are used, as they are in the method of Kocher:

> . . .[It is possible that] there is no storage of random numbers Z and function value f(Z) since they are generated by means of a suitable generator whenever required. It is important that the generator or generators do not generate function values f(Z) by applying linear function f to random number Z but that pairs of random numbers Z and function values f(Z) be generated in another way since random number Z may otherwise be spied out by interception of the application of the function f to random number Z and further secret data determined with the air of this information. . ..

In contrast, Kocher's method of computing the auxiliary function value by applying the permutation given by "table" (see the last line of the third for-loop in the source of paragraph [0068]), is exactly the type of non-previous computation warned about in Applicant's original specification.

In summary, because the approach of Kocher is to falsify the input data by auxiliary data *generated during the falsification procedure* and then to compensate for that falsification by combining the result with auxiliary function values *generated during execution of the operation*, rather than being generated previously in safe surroundings and stored, the subject matter of claim 26 is not anticipated or rendered obvious by the Kocher publication, and withdrawal of the rejection of claims 26-33 and 42 is respectfully requested.

Having thus overcome the sole rejection made in the Official Action, expedited passage of the application to issue is requested.

Respectfully submitted,

BACON & THOMAS, PLLC

Date: October 19, 2007

By: BENJAMIN E. URCIA
Registration No. 33,805

Serial Number **09/700,656**

BACON & THOMAS, PLLC
625 Slaters Lane, 4th Floor
Alexandria, Virginia 22314

Telephone: (703) 683-0500

NWB:S:\Producer\beu\Pending Q...Z\V\VATER 700656\a05.wpd